



嵌入式系统

Embedded System

毛维杰

杭州 • 浙江大学 • 2021

第四章 组合逻辑电路

4.1 概述

数字逻辑电路 { 组合逻辑电路
时序逻辑电路

组合电路特点 { 功能上: 输出仅与该时刻的输入有关。
结构上: 由门电路组成, 不含记忆
(存储) 元件。

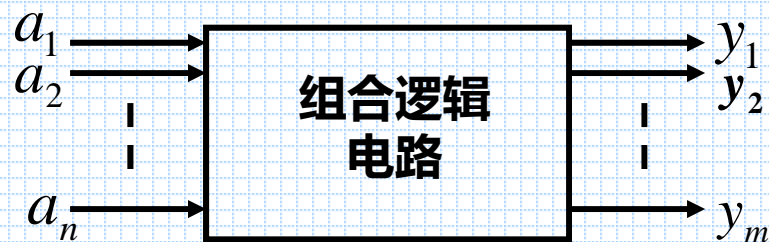
主要内容

- 1. 组合电路的分析、设计方法。
- 2. 常用集成组合逻辑电路。

由小规模集成电路组成

由中规模集成电路组成

二、逻辑功能的描述



组合逻辑电路的框图

$$y_1 = f_1(a_1 a_2 \cdots a_n)$$

$$y_2 = f_2(a_1 a_2 \cdots a_n)$$

\vdots

$$y_m = f_m(a_1 a_2 \cdots a_n)$$

$$Y = F(A)$$

4.2 组合逻辑电路的设计方法

一、逻辑抽象

- 分析因果关系，确定输入/输出变量
- 定义逻辑状态的含意（赋值）
- 列出真值表

二、写出函数式

三、选定器件类型

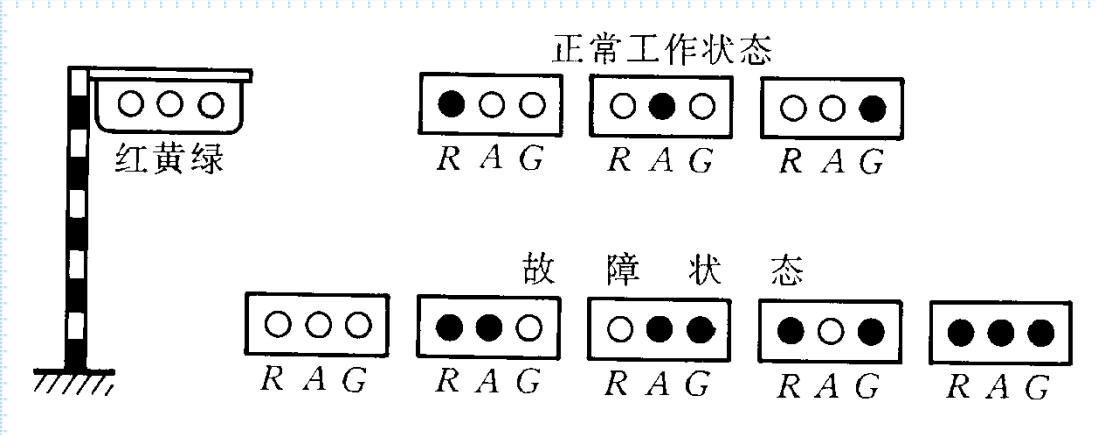
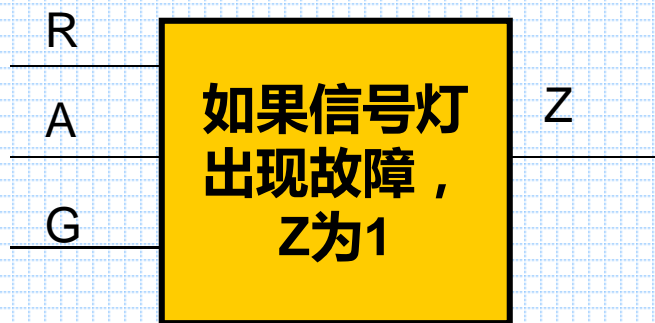
- ### 四、根据所选器件：
- 对逻辑式化简（用门）
 - 变换（用MSI）
 - 或进行相应的描述（PLD）

五、画出逻辑电路图，或下载到PLD

六、工艺设计

设计举例：

- 设计一个监视交通信号灯状态的逻辑电路



设计举例：

1. 抽象

- 输入变量：

红 (R)、黄 (A)、绿 (G)

- 输出变量：

故障信号 (Z)

2. 写出逻辑表达式

$$Z = R'A'G' + R'AG + RA'G + RAG' + RAG$$

输入变量			输出
R	A	G	Z
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

设计举例：

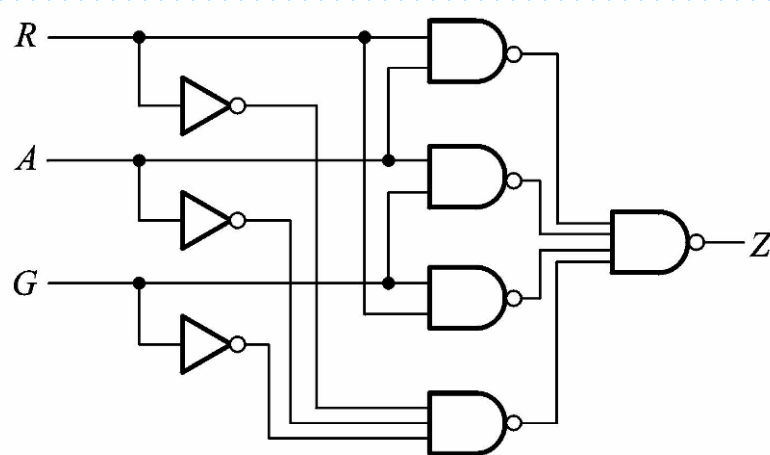
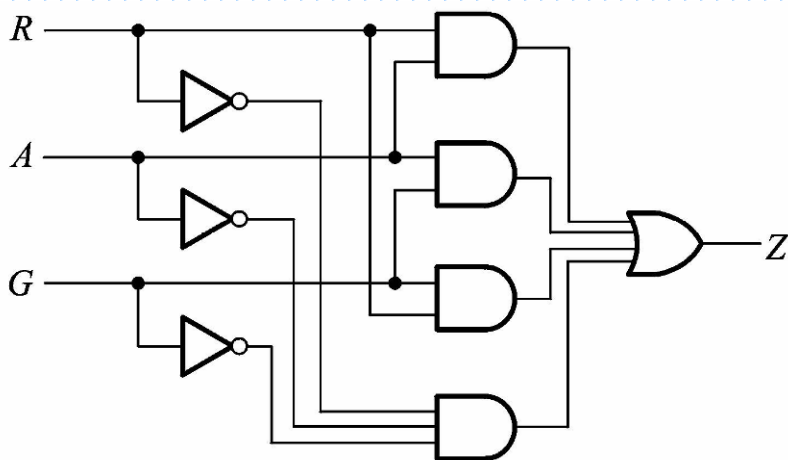
3. 选用小规模SSI器件

4. 化简

$$Z = R' A' G' + RA + RG + AG$$

5. 画出逻辑图

		AG			
		00	01	11	10
R	0	1	0	1	0
	1	0	1	1	1



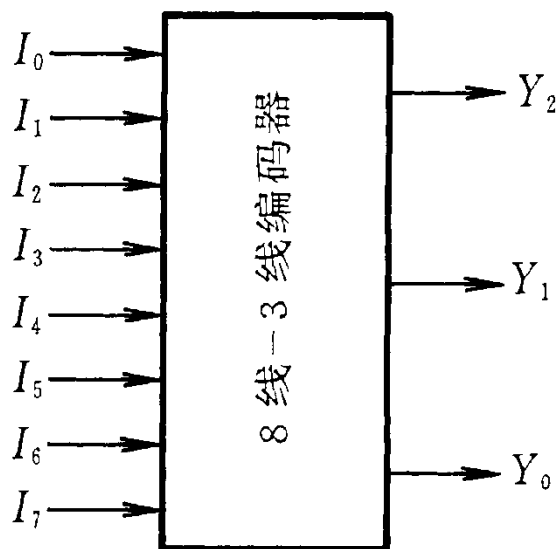
4.3 若干常用组合逻辑电路

4.3.1 编码器

- 编码：将输入的每个高/低电平信号变成一个对应的二进制代码
- 普通编码器
- 优先编码器

一、普通编码器

- 特点：任何时刻只允许输入一个编码信号。
- 例：3位二进制普通编码器



输 入								输 出		
I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	Y_2	Y_1	Y_0
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

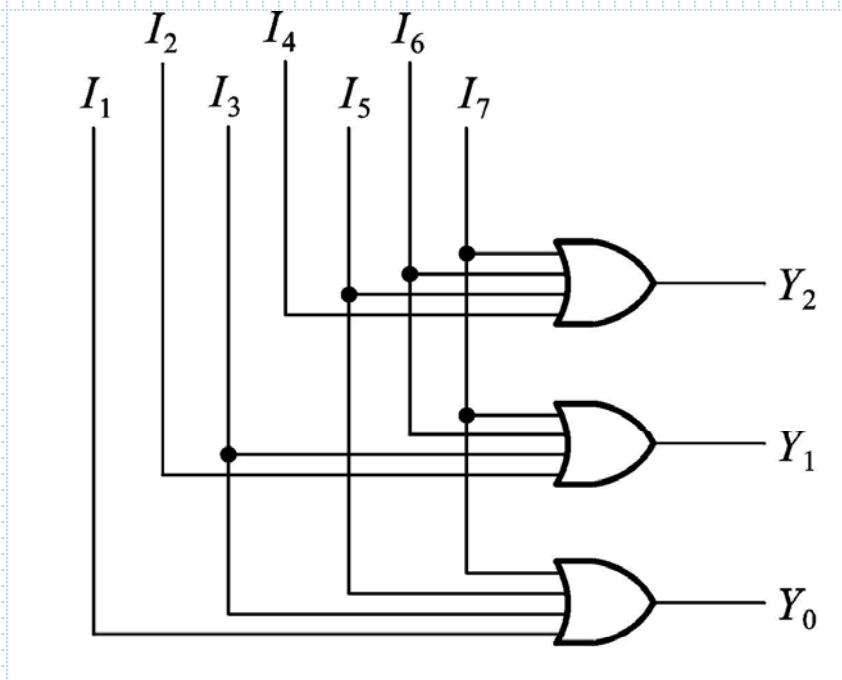
$$\begin{aligned}
 Y_2 = & I_7' I_6' I_5' I_4' I_3' I_2' I_1' I_0' + I_7' I_6' I_5' I_4' I_3' I_2' I_1' I_0 \\
 & + I_7' I_6' I_5' I_4' I_3' I_2' I_1' I_0' + I_7' I_6' I_5' I_4' I_3' I_2' I_1' I_0
 \end{aligned}$$

利用无关项化简，得：

$$Y_2 = I_4 + I_5 + I_6 + I_7$$

$$Y_1 = I_2 + I_3 + I_6 + I_7$$

$$Y_0 = I_1 + I_3 + I_5 + I_7$$

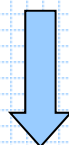


二、优先编码器

- 特点：允许同时输入两个以上的编码信号，但只对其中优先权最高的一个进行编码。
- 例：8线-3线优先编码器
- （设 I_7 优先权最高... I_0 优先权最低）

输 入								输 出		
I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	Y_2	Y_1	Y_0
X	X	X	X	X	X	X	1	1	1	1
X	X	X	X	X	X	1	0	1	1	0
X	X	X	X	X	1	0	0	1	0	1
X	X	X	X	1	0	0	0	1	0	0
X	X	X	1	0	0	0	0	0	1	1
X	X	1	0	0	0	0	0	0	1	0
X	1	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0

$$Y_2 = I_7 + I_7' I_6 + I_7' I_6' I_5 + I_7' I_6' I_5' I_4$$



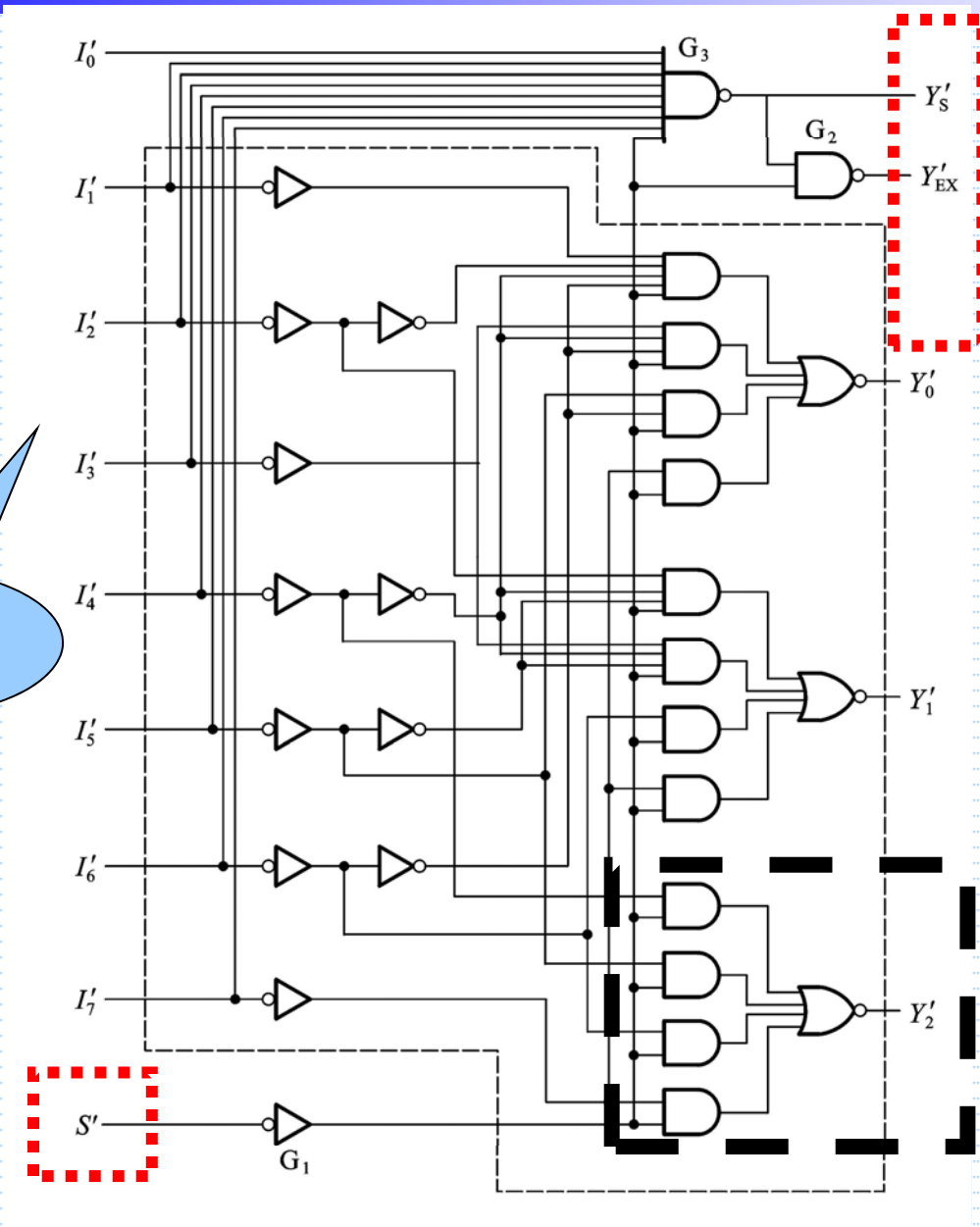
$$A + A'B = A + B$$

$$Y_2 = I_7 + I_6 + I_5 + I_4$$

实例： 74HC148

低电平

选通信号



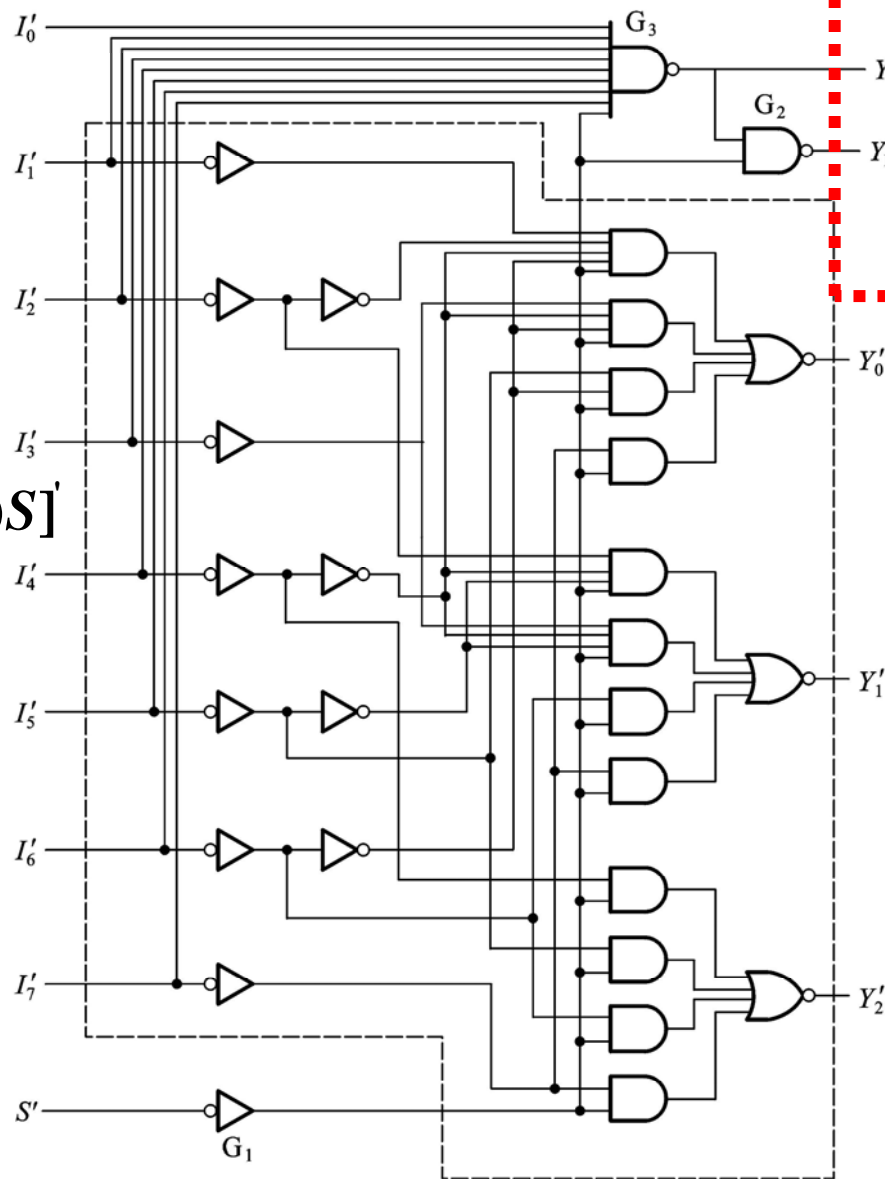
为0时，电路工作
无编码输入

$$Y'_S = (I'_7 I'_6 I'_5 I'_4 I'_3 I'_2 I'_1 I'_0 S')$$

$$Y'_{EX} = [(I'_7 I'_6 I'_5 I'_4 I'_3 I'_2 I'_1 I'_0 S') S']$$

$$[(I_7 + I_6 + I_5 + I_4 + I_3 + I_2 + I_1 + I_0) S']$$

为0时，电路工作
有编码输入



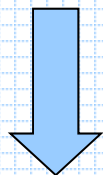
[illegible]

附加输出信号的状态及含意

Y'_S	Y'_{EX}	状态
1	1	不工作
0	1	工作，但无输入
1	0	工作，且有输入
0	0	不可能出现

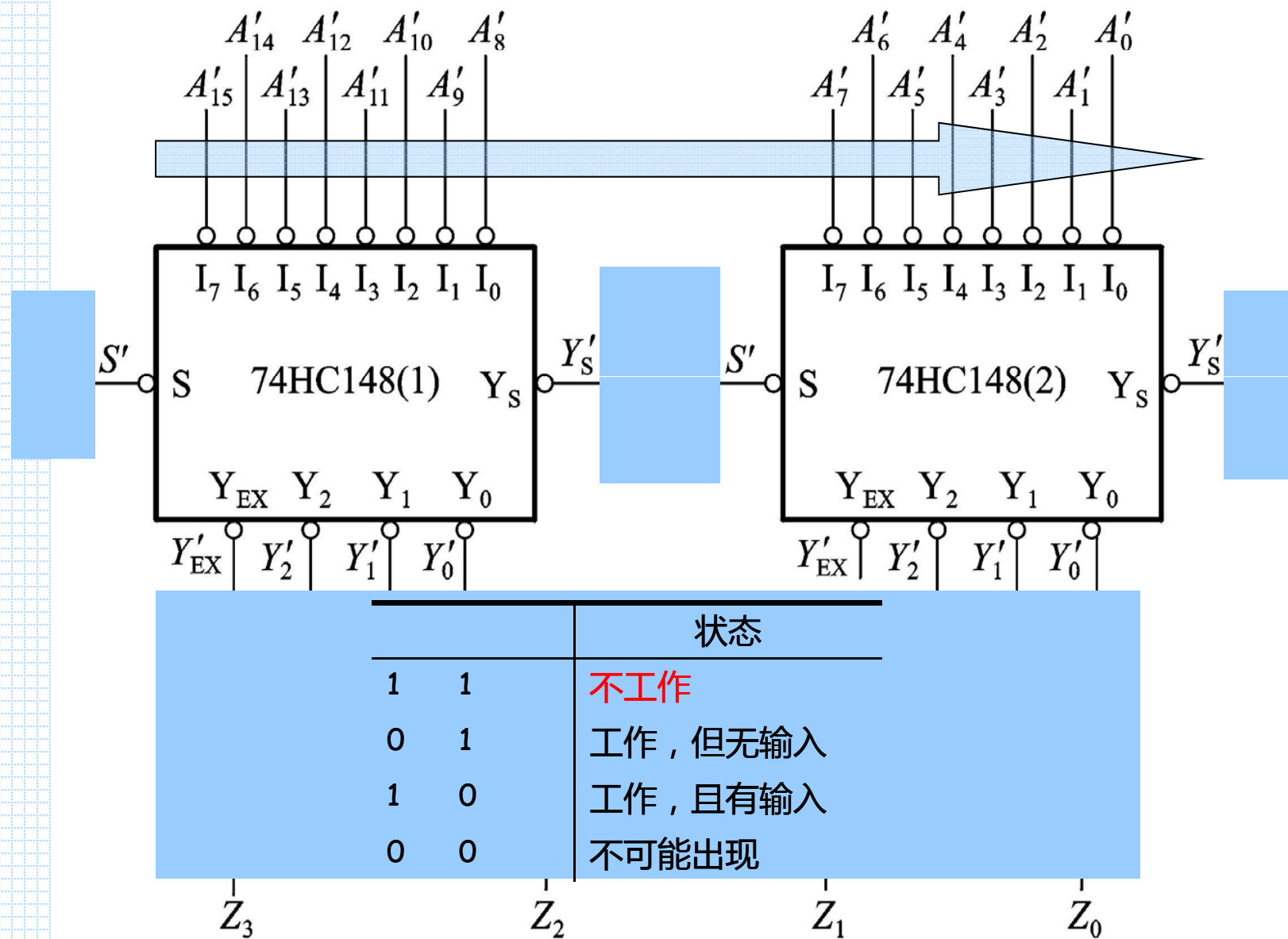
控制端扩展功能举例：

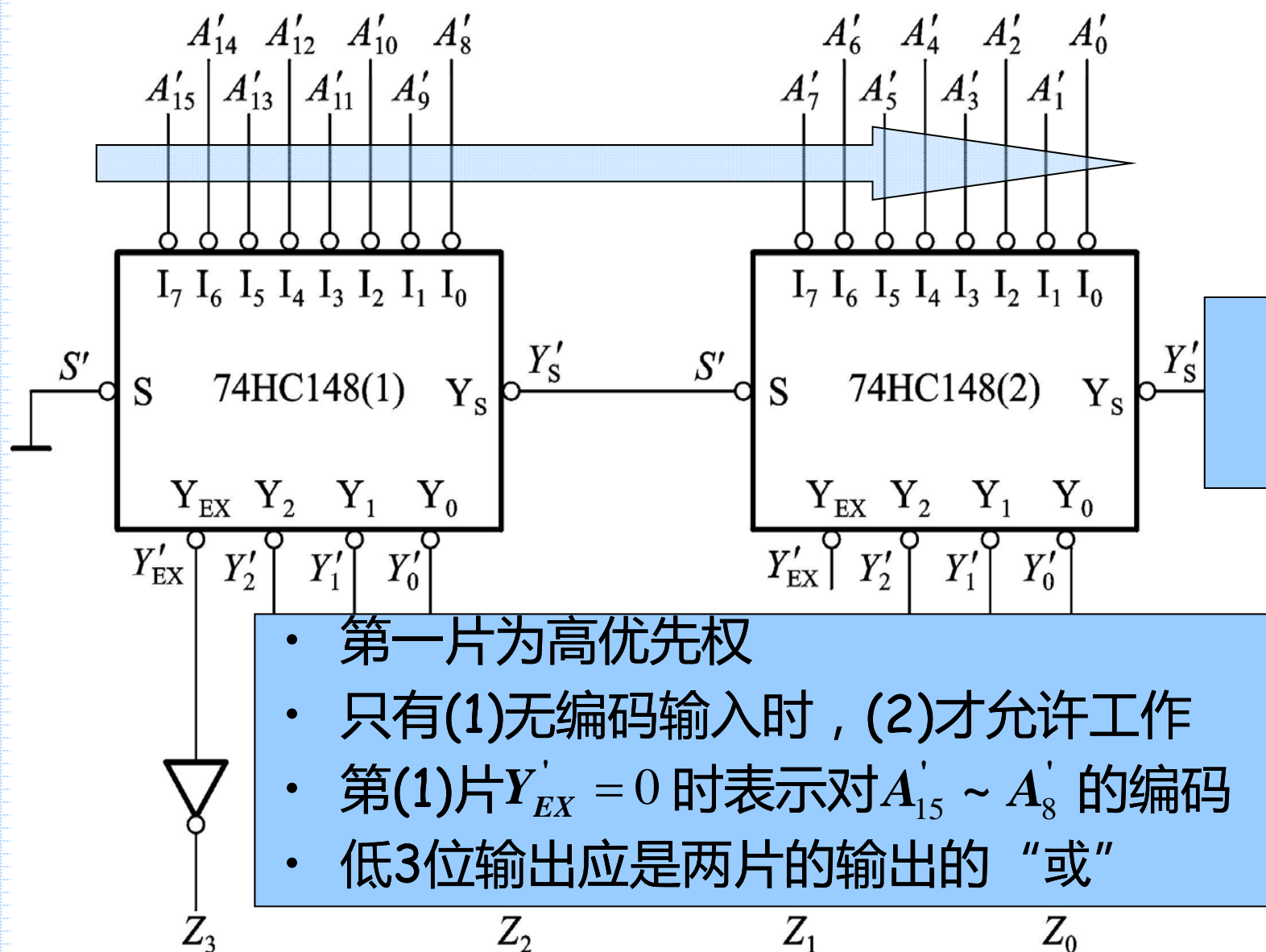
- 例： 用两片8线-3线优先编码器

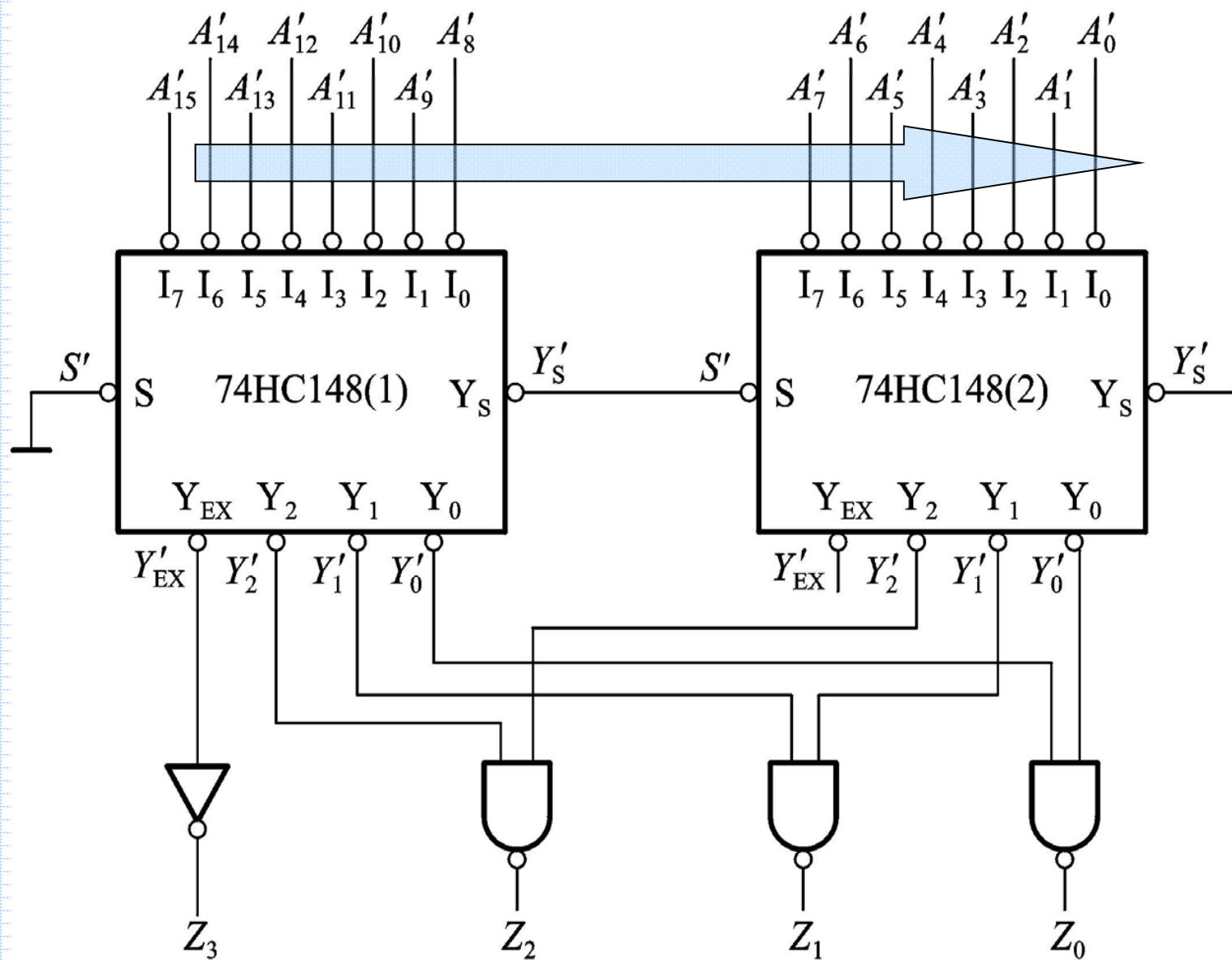


16线-4线优先编码器

其中， A'_{15} 的优先权最高...





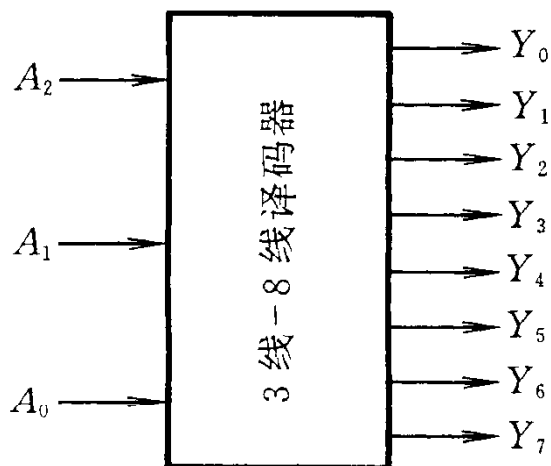


4.3.2 译码器

- 译码：将每个输入的二进制代码译成对应的输出高、低电平信号。
- 常用的有：二进制译码器，二-十进制译码器，显示译码器等

一、二进制译码器

例：3线—8线译码器

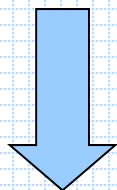


输 入			输 出							
A_2	A_1	A_0	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

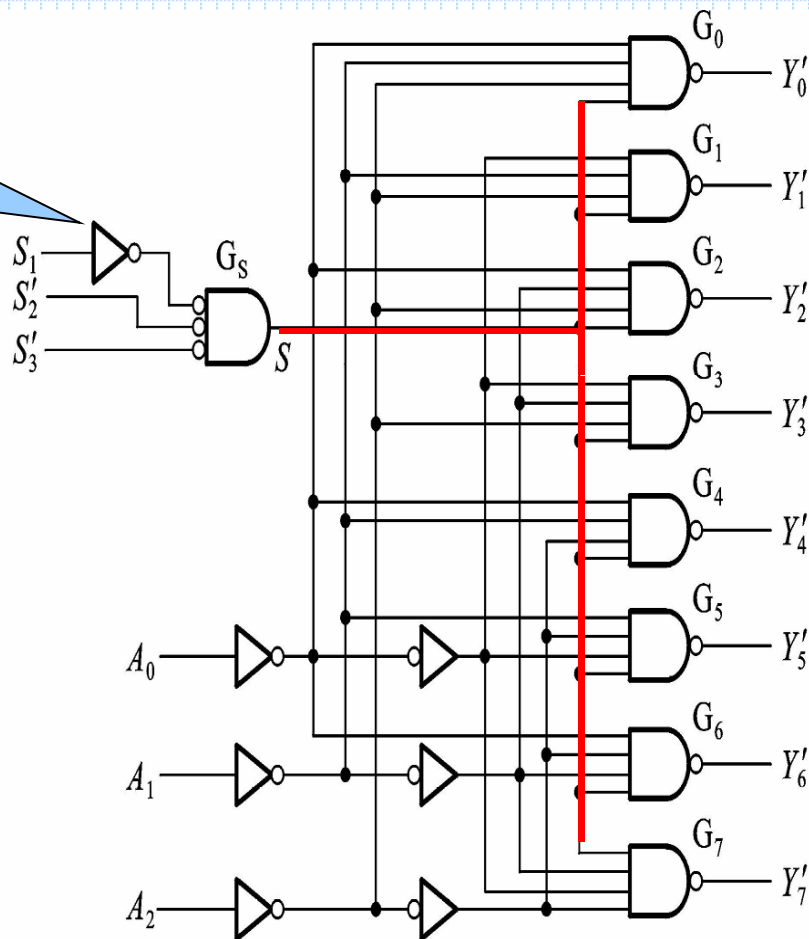
集成译码器实例：74HC138

附加
控制端

$$S = S_3 S_2 S_1$$



$$Y_i' = (S \ m_i)'$$

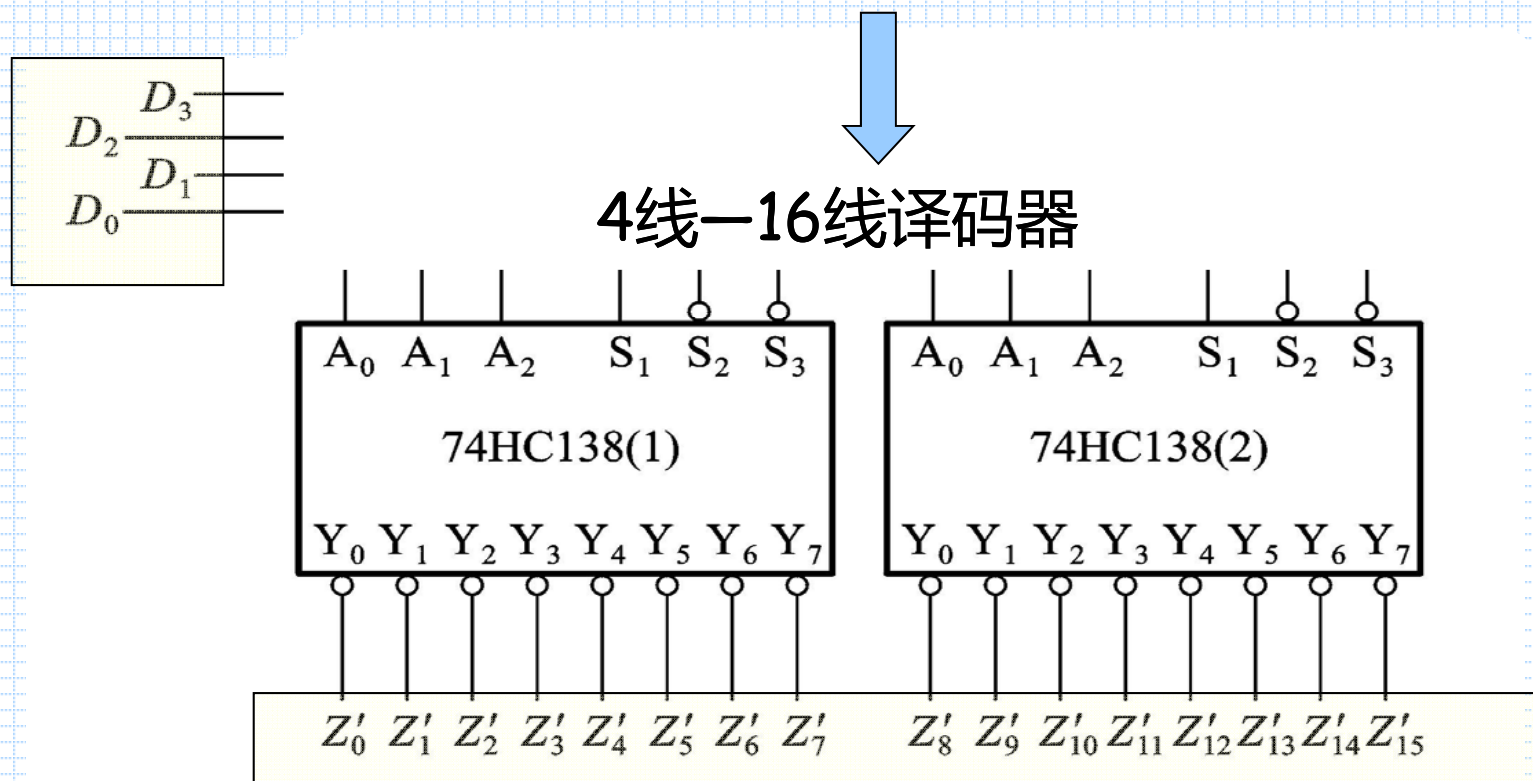
低电平
输出

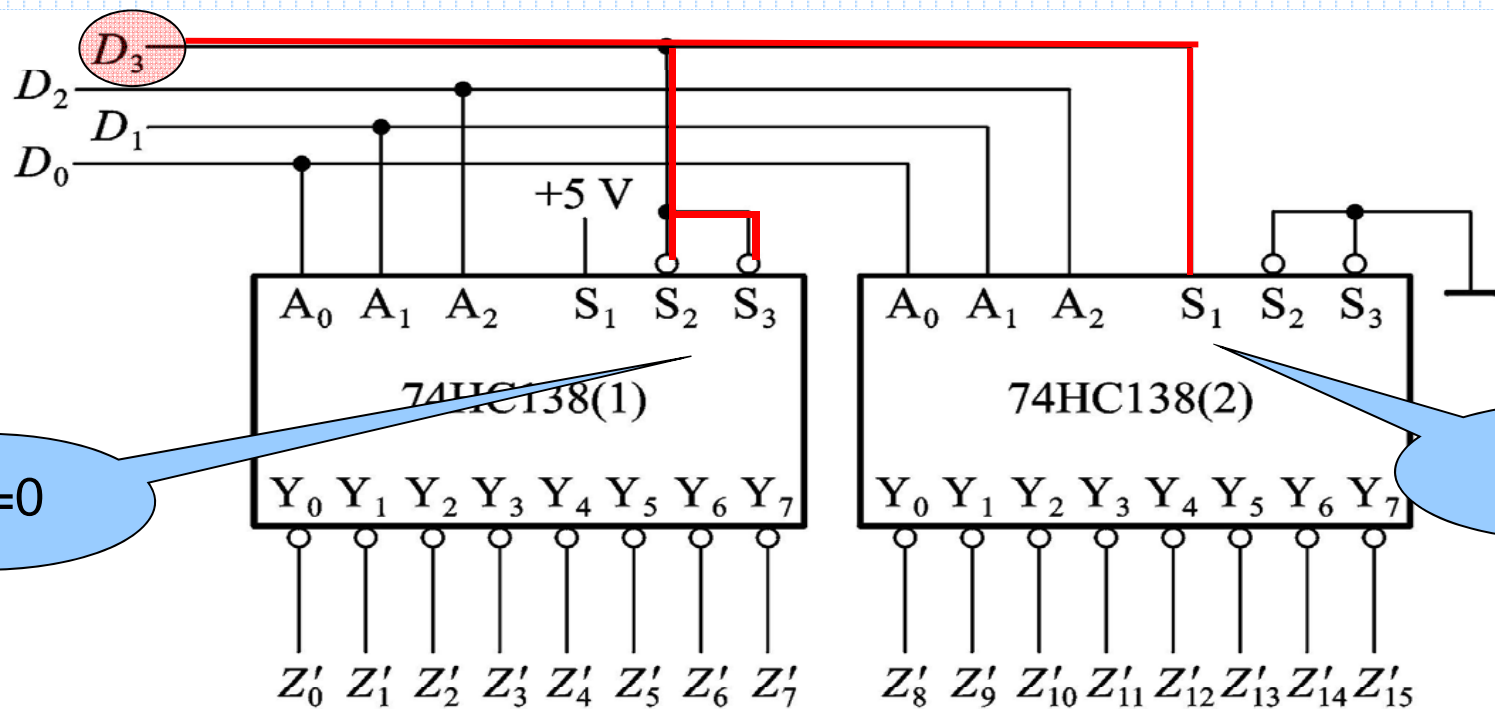
74HC138的功能表：

输 入					输 出							
S_1	$S_2' + S_3'$	A_2	A_1	A_0	Y_7'	Y_6'	Y_5'	Y_4'	Y_3'	Y_2'	Y_1'	Y_0'
0	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	1	1	1	1	1	1	1	1
1	0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	0	1	1	1	1	1	1	1	0	1
1	0	0	1	0	1	1	1	1	1	0	1	1
1	0	0	1	1	1	1	1	1	0	1	1	1
1	0	1	0	0	1	1	1	0	1	1	1	1
1	0	1	0	1	1	1	0	1	1	1	1	1
1	0	1	1	0	1	0	1	1	1	1	1	1
1	0	1	1	1	0	1	1	1	1	1	1	1

- 利用附加控制端进行扩展

例： 用74HC138（3线—8线译码器）





$D_3=0$

$D_3=1$

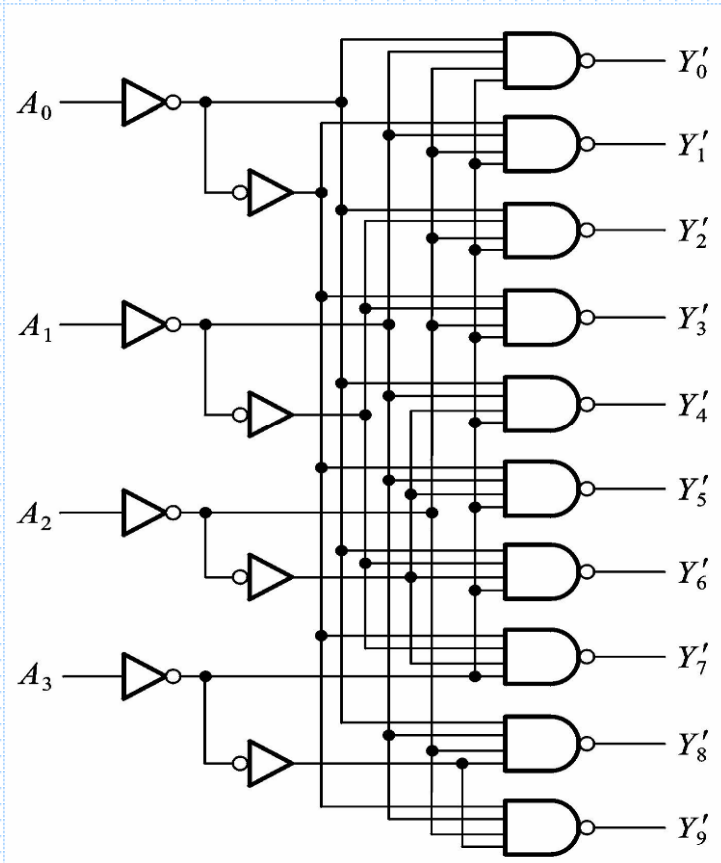
$$Z'_i = m'_i$$

二、二—十进制译码器

- 将输入BCD码的10个代码译成10个高、低电平的输出信号
BCD码以外的伪码，输出均无低电平信号产生

- 例：74HC42

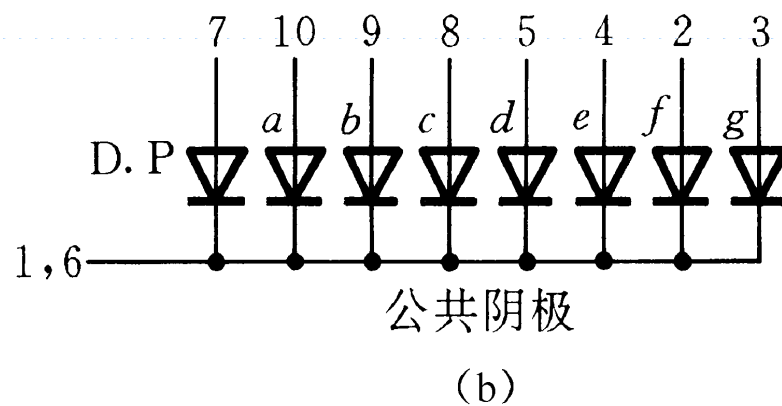
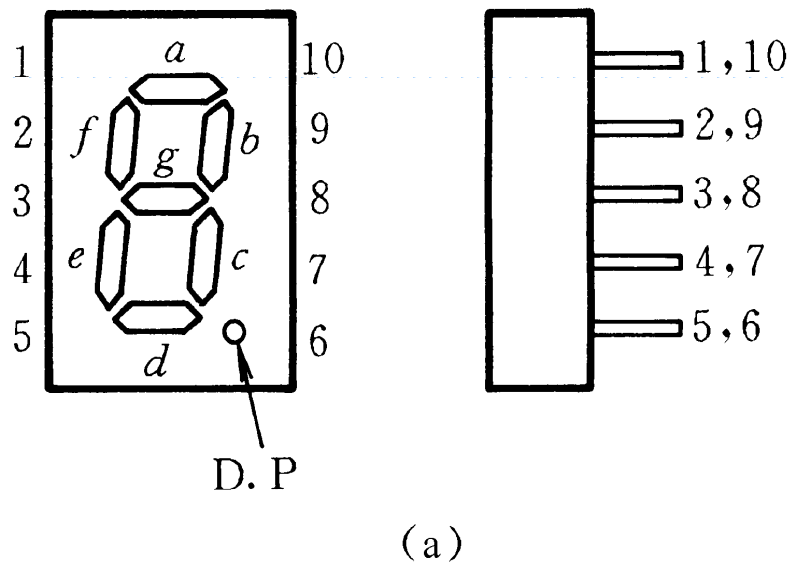
$$Y'_i = m'_i \quad (i = 0 \sim 9)$$



三、显示译码器

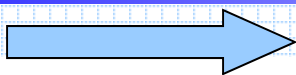
• 1. 七段字符显示器

如：



(代码转换器) 7448

真值表



卡诺图

$A_3A_2 \backslash A_1A_2$		A_1A_2			
		00	01	11	10
00	1	0	1	1	
01	0	1	1	0	
11	0	1	0	0	
10	1	1	0	0	

(a)

$A_3A_2 \backslash A_1A_2$		A_1A_2			
		00	01	11	10
00	1	1	1	1	
01	1	0	1	0	
11	1	0	0	0	
10	1	1	0	0	

(b)

$A_3A_2 \backslash A_1A_2$		A_1A_2			
		00	01	11	10
A_3A_2	00	1	1	1	0
	01	1	1	1	1
	11	0	0	0	0
	10	1	1	1	0

(c)

$A_3A_2 \backslash A_1A_2$		A_1A_2			
		00	01	11	10
A_3A_2	00	1	0	1	1
	01	0	1	0	1
	11	0	1	0	1
	10	1	0	1	1

(d)

$A_3A_2 \backslash A_1A_2$		A_1A_2			
		00	01	11	10
A_3A_2	00	1	0	0	1
	01	0	0	0	1
	11	0	0	0	1
	10	1	0	0	1

(e)

$A_3A_2 \backslash A_1A_2$		A_1A_2			
		00	01	11	10
00	1	0	0	0	
01	1	1	0	1	
11	1	1	0	1	
10	1	1	0	0	

(f)

$A_3 A_2 \backslash A_1 A_2$		$A_1 A_2$			
		00	01	11	10
00	0	0	1	1	
01	1	1	0	1	
11	1	1	0	1	
10	1	1	1	1	

(g)

BCD - 七段显示译码器7448的逻辑图

$$Y_a = (A_3' A_2' A_1' A_0 + A_3 A_1 + A_2 A_0)'$$

$$Y_b = (A_3 A_1 + A_2 A_1 A_0' + A_2 A_1' A_0)'$$

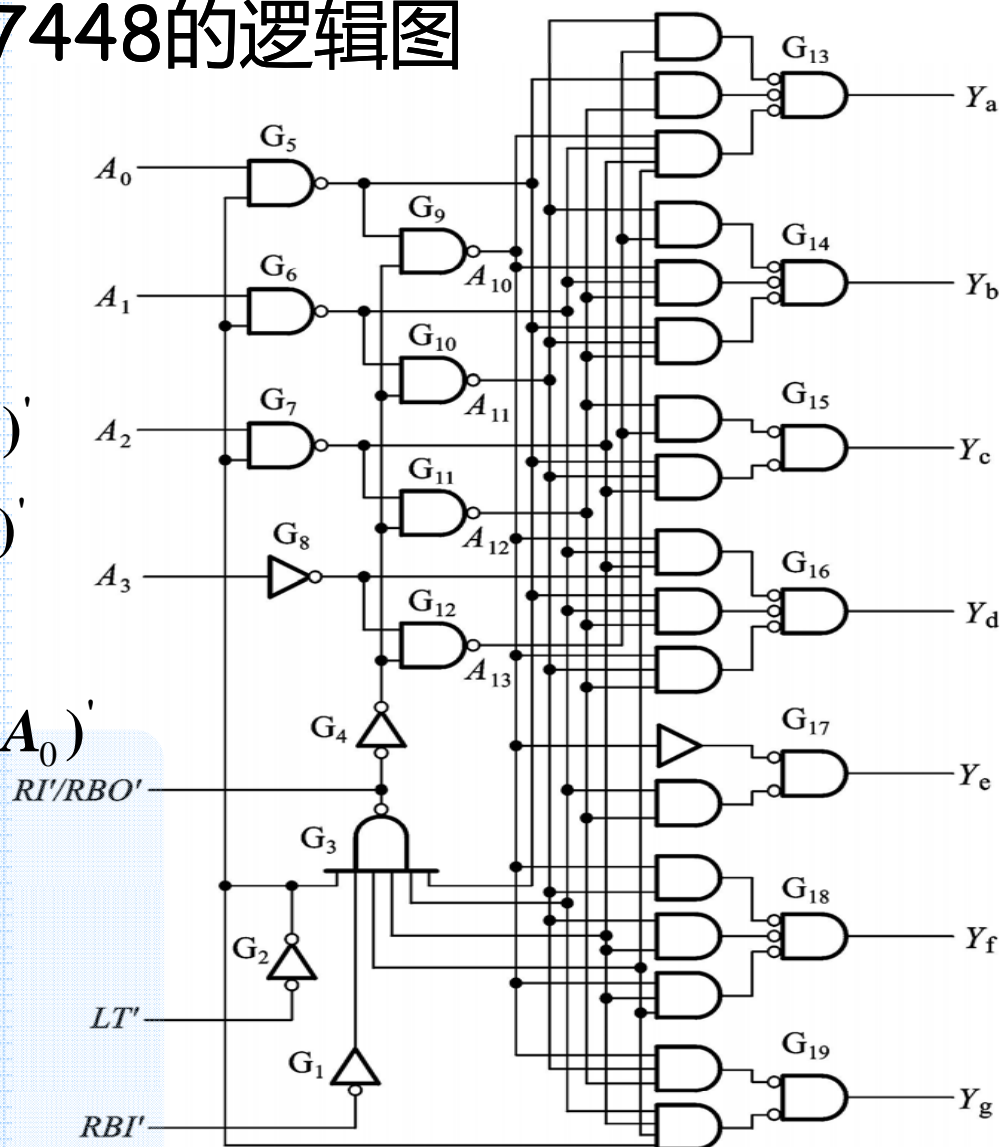
$$Y_c = (A_3 A_2 + A_2' A_1 A_0)'$$

$$Y_d = (A_2 A_1 A_0 + A_2 A_1' A_0' + A_2' A_1' A_0)'$$

$$Y_e = (A_2 A_1' + A_0)'$$

$$Y_f = (A_3' A_2' A_0 + A_2' A_1 + A_1 A_0)'$$

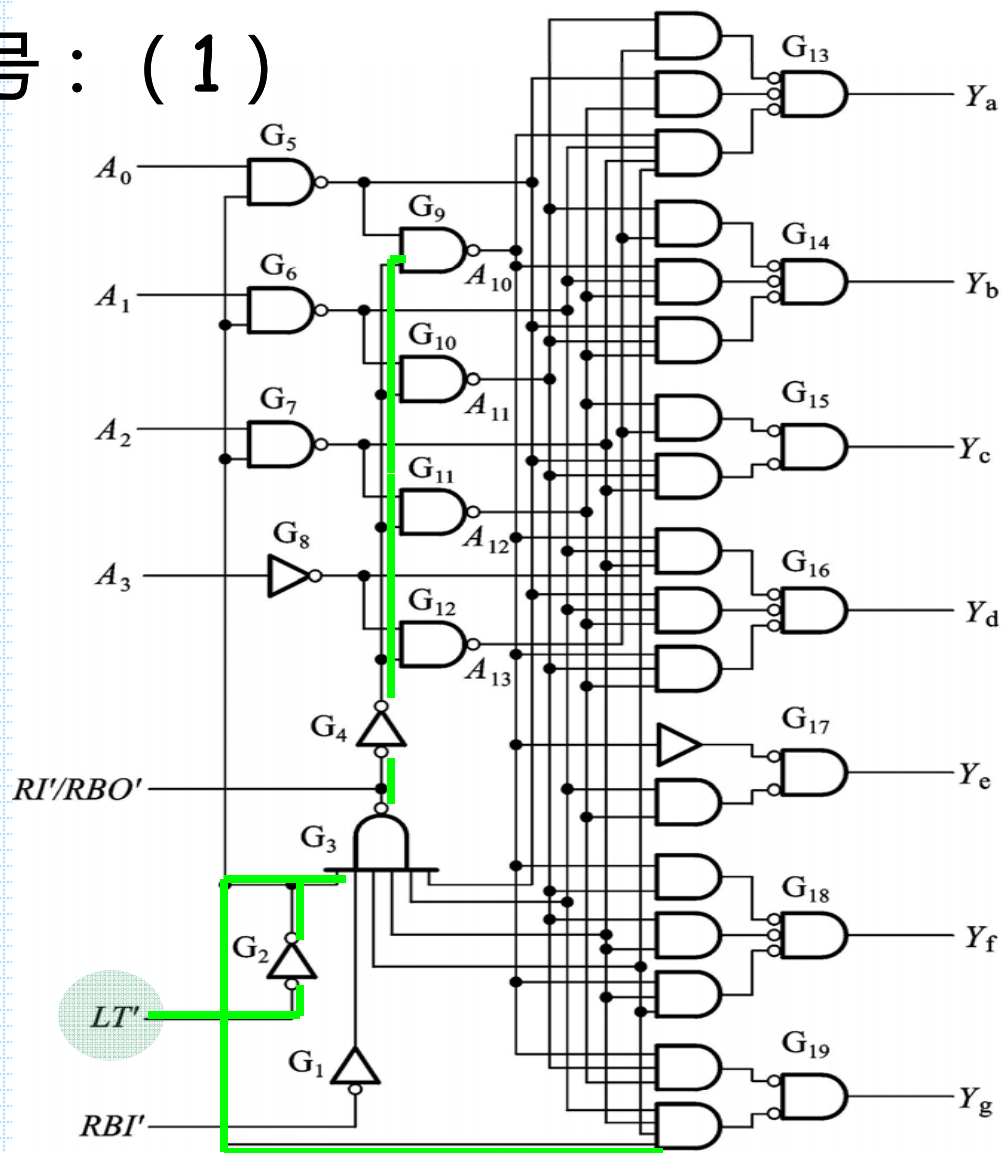
$$Y_g = (A_3' A_2' A_1' + A_2 A_1 A_0)'$$



7448的附加控制信号：(1)

- 灯测试输入 LT'

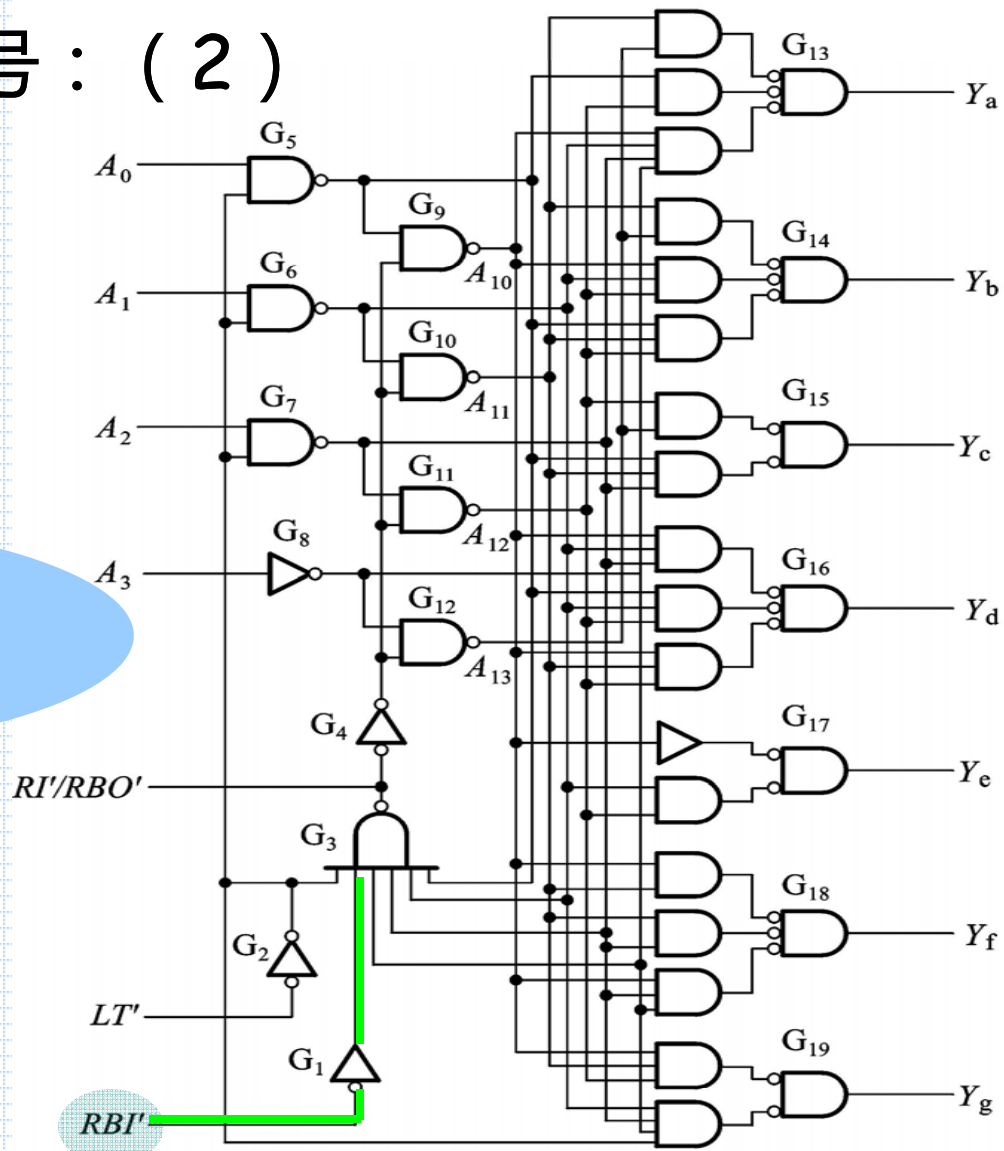
当 $LT' = 0$ 时, $Y_a \sim Y_g$ 全部置为1



7448的附加控制信号：(2)

- 灭零输入 RBI'

当 $A_3A_2A_1A_0 = 0000$ 时，
 $RBI' = 0$ 时，则灭灯



7448的附加控制信号：（3）

- 灭灯输入/灭零输出 BI'/RBO'

输入信号，称灭灯输入控制端：

$BI' = 0$ 无论输入状态是什么，数码管熄灭

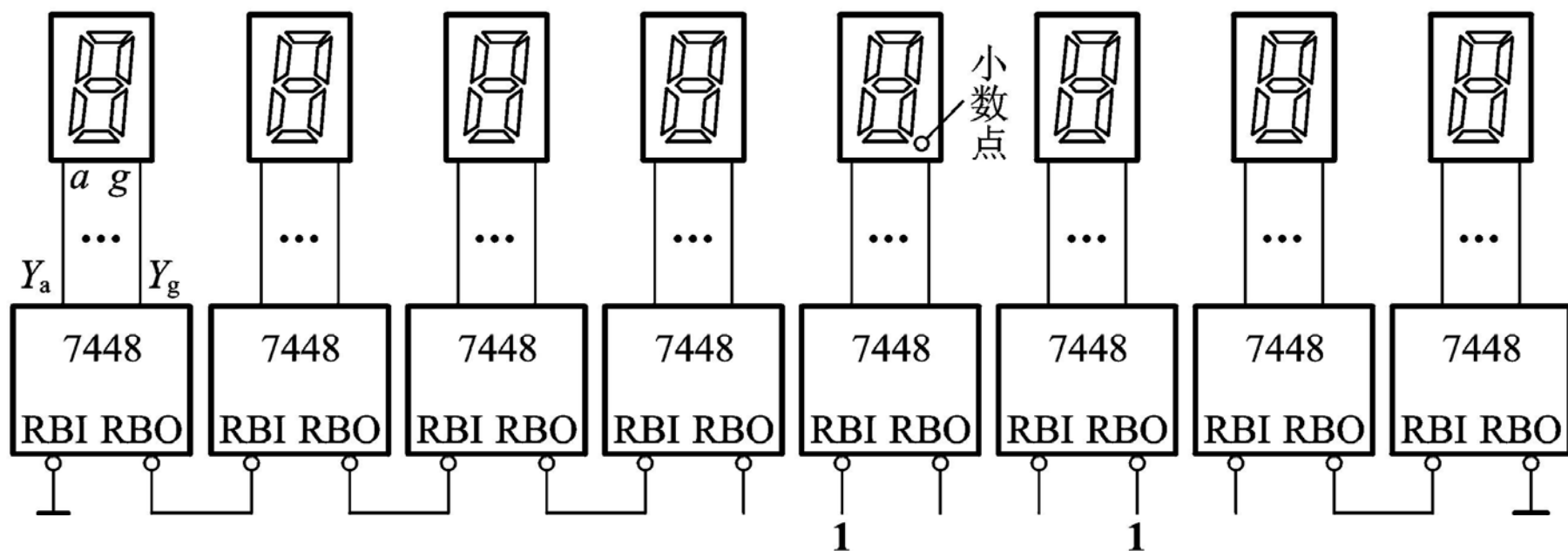
输出信号，称灭零输出端：

只有当输入 $A_3A_2A_1A_0 = 0$ ，且灭零输入信号 $RBI' = 0$ 时，
 RBO' 才给出低电平

因此 $RBO' = 0$ 表示译码器将本来应该显示的零熄灭了

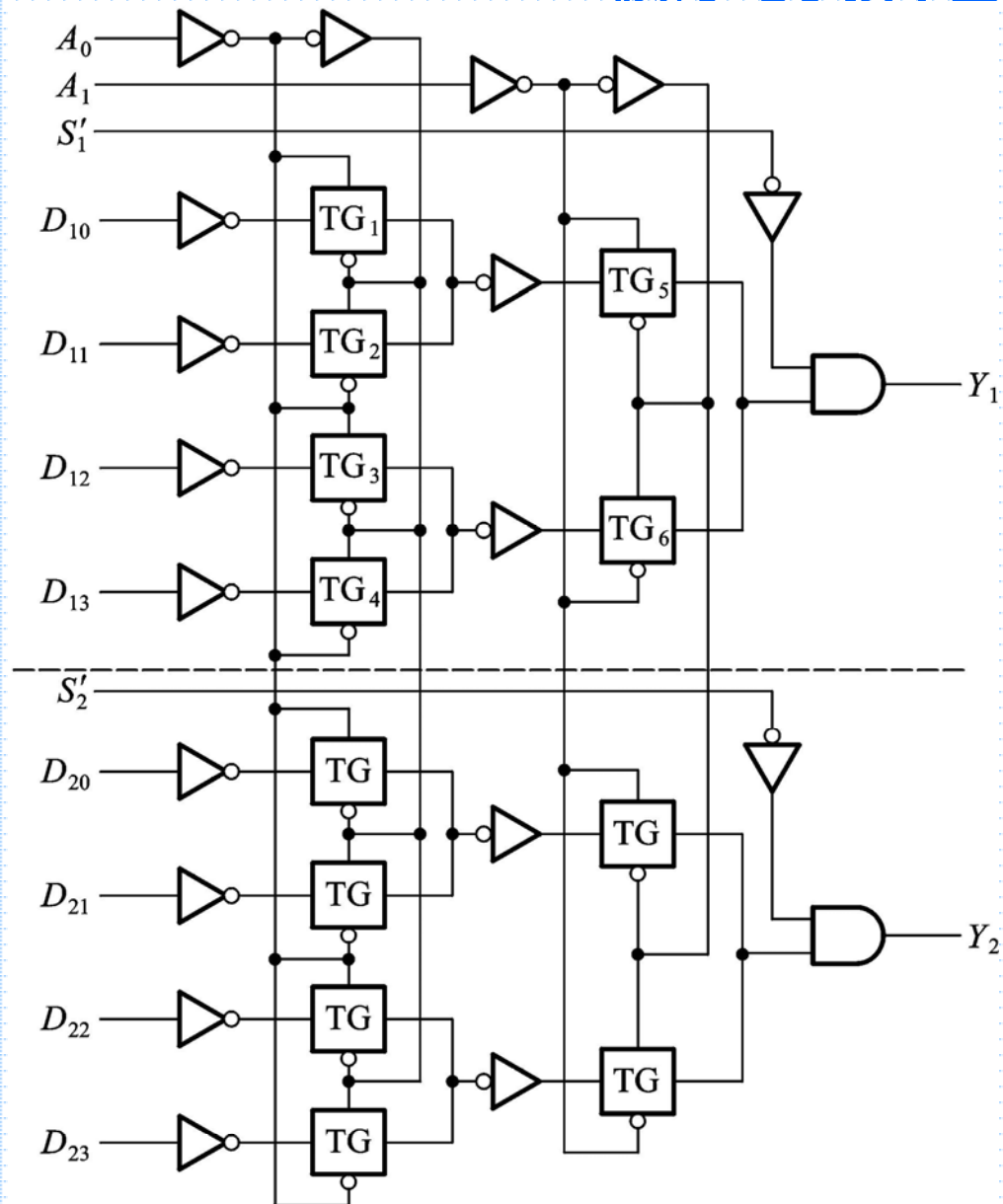
例：利用 RBI' 和 RBO' 的配合，实现多位显示系统的灭零控制

- 整数部分：最高位是0，而且灭掉以后，输出 RBO' 作为次高位的输入信号 RBI'
- 小数部分：最低位是0，而且灭掉以后，输出 RBO' 作为次低位的输入信号 RBI'

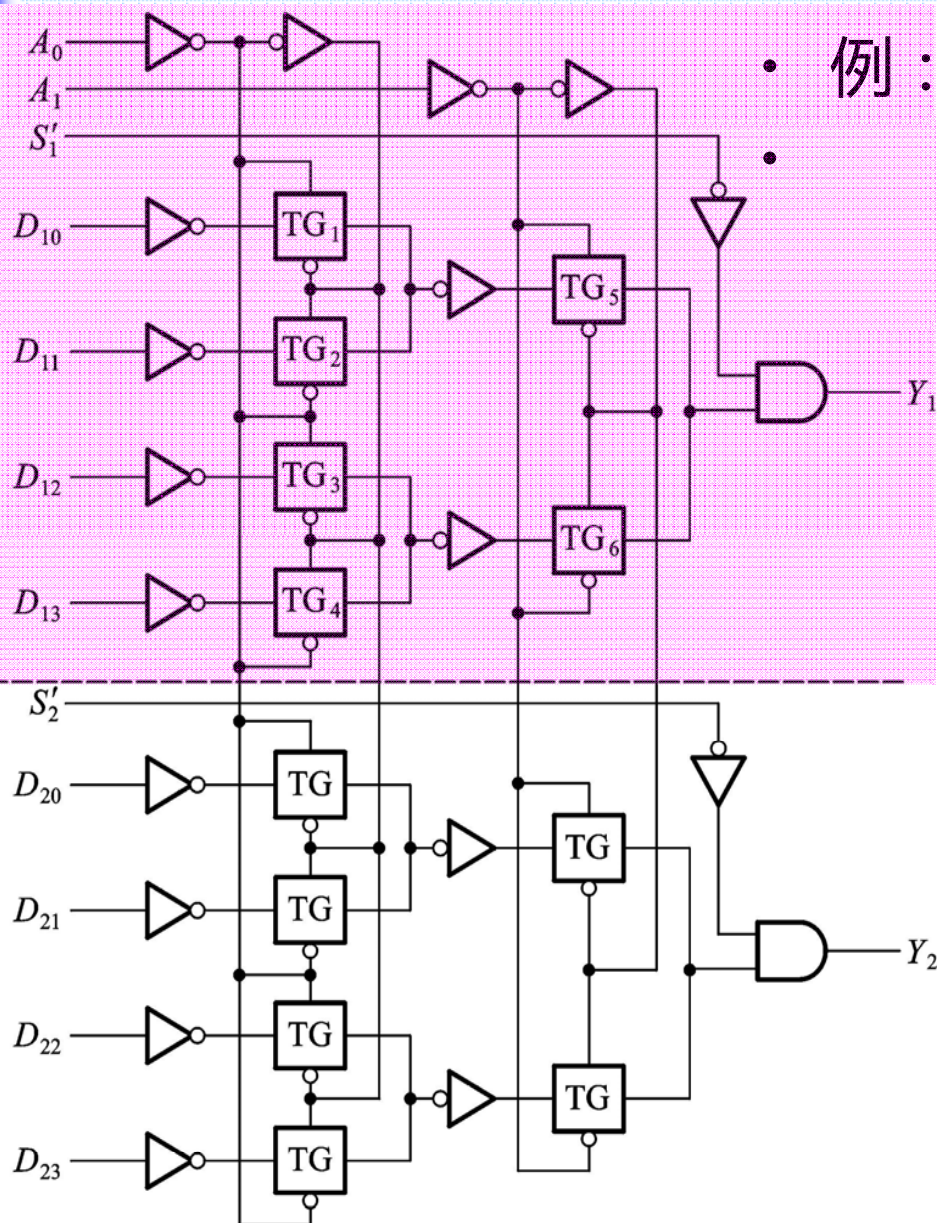


4.3.3 数据选择器

一、工作原理



例：“双四选一”，74HC153
分析其中的一个“四选一”

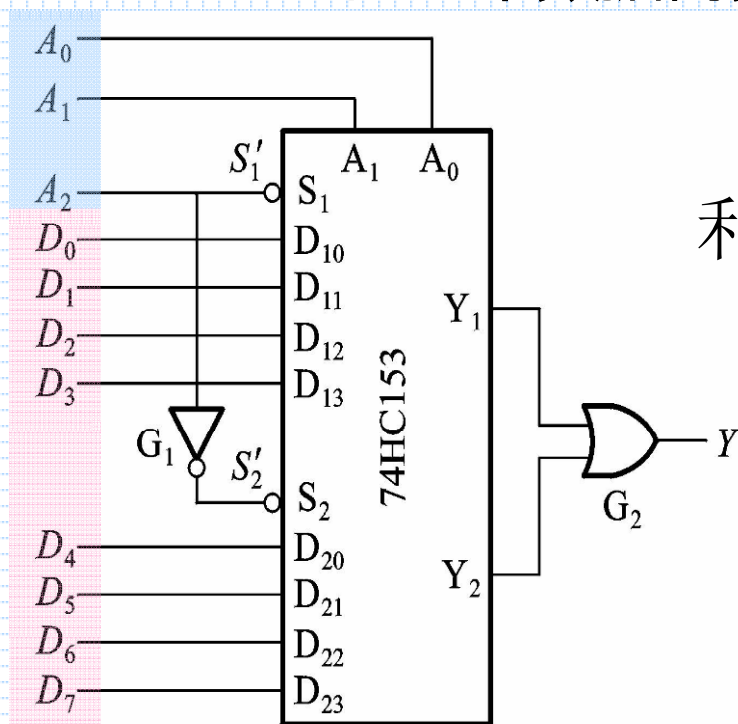


$$Y_1 = S_1' [D_0(A_1'A_0') + D_1(A_1'A_0) + D_2(A_1A_0') + D_3(A_1A_0)]$$

S_1'	A_1	A_0	Y_1
1	X	X	0
0	0	0	D_{10}
0	0	1	D_{11}
0	1	0	D_{12}
0	1	1	D_{13}

例：用两个“四选一”接成“八选一”

- “四选一” 只有2位地址输入，从四个输入中选中一个
- “八选一” 的八个数据需要3位地址代码指定其中任何一个



利用 S' 作为第 3 位地址输入端

$$Y = (A_2' A_1' A_0') D_0 + (A_2' A_1' A_0) D_1 + (A_2' A_1 A_0') D_2 + (A_2' A_1 A_0) D_3 \\ + (A_2 A_1' A_0') D_4 + (A_2 A_1' A_0) D_5 + (A_2 A_1 A_0') D_6 + (A_2 A_1 A_0) D_7$$

4.3.4 加法器

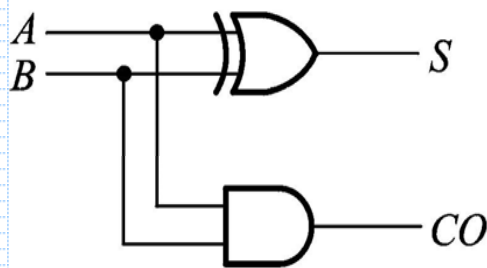
一、1位加法器

1. 半加器，不考虑来自低位的进位，将两个1位的二进制数相加

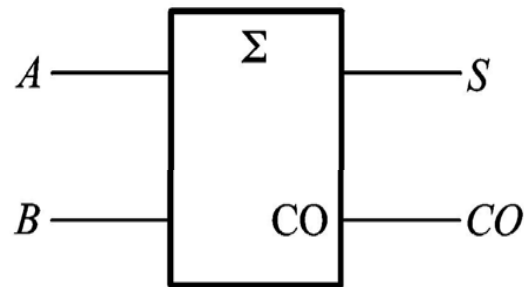
输 入		输 出	
A	B	S	CO
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$S = A \oplus B$$

$$CO = AB$$



(a)



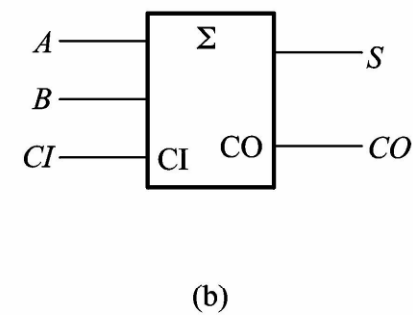
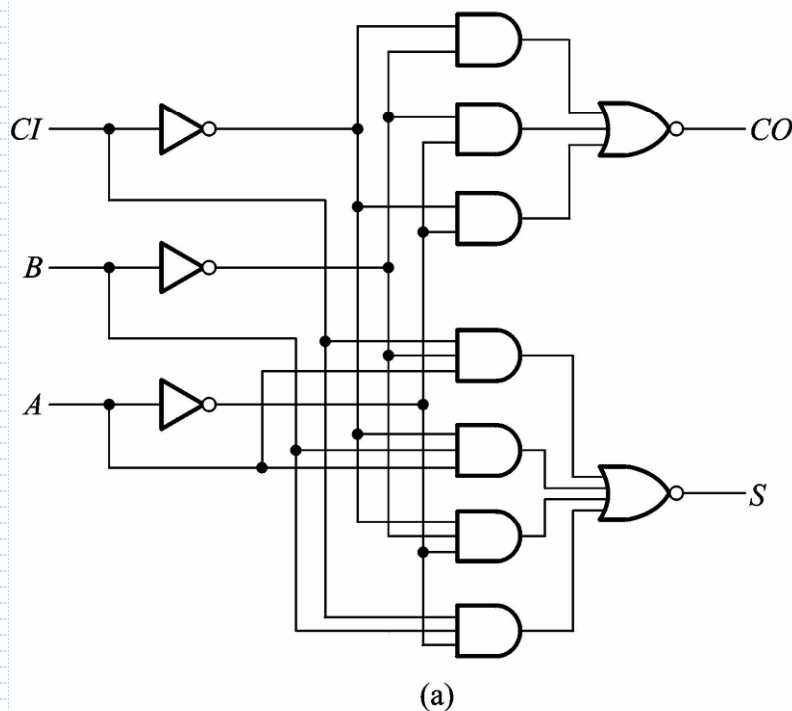
(b)

2. 全加器：将两个1位二进制数及来自低位的进位相加

输 入			输 出	
A	B	CI	S	CO
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S = (A'B'CI' + A'B \cdot CI + AB'CI + ABCI')$$

$$CO = (A'B' + B'CI' + A'CI')$$



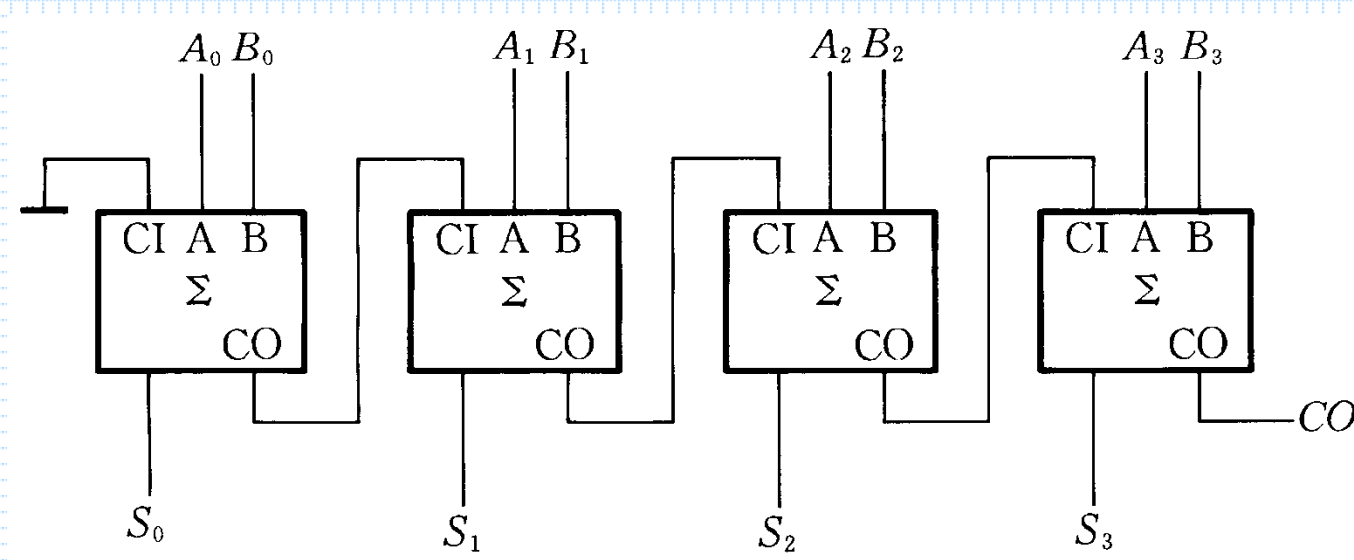
74HC183

二、多位加法器

1. 串行进位加法器

优点：简单

缺点：慢



$$(CI)_i = (CO)_{i-1}$$

$$S_i = A_i \oplus B_i \oplus (CI)_i$$

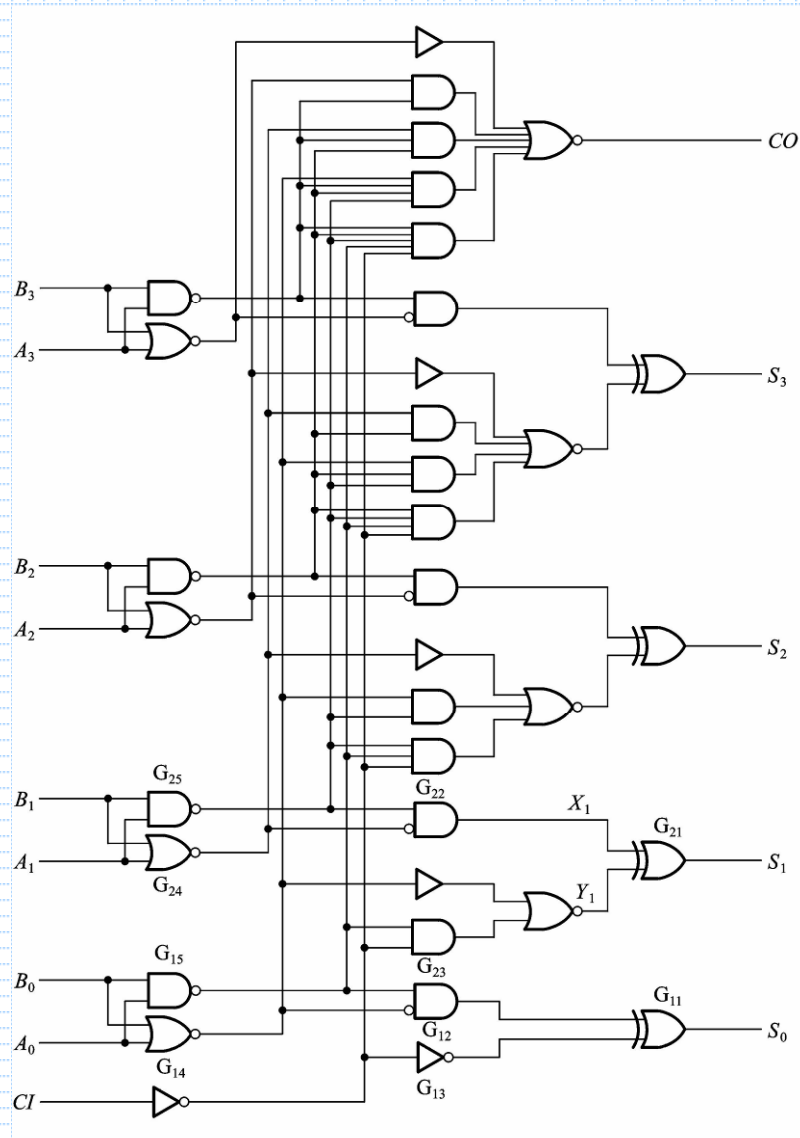
$$(CO)_i = A_i B_i + (A_i + B_i)(CI)_i$$

2. 超前进位加法器

基本原理：加到第 i 位的进位输入信号是两个加数第 i 位以前各位（ $0 \sim j-1$ ）的函数，可在相加前由 A, B 两数确定。

优点：快，每1位的和及最后的进位基本同时产生。

缺点：电路复杂。



74LS283

$$i = 0: (CI)_0 = 0$$

$$S_0 = A_0 \oplus B_0 \oplus (CI)_0$$

$$(CO)_0 = A_0 B_0 + (A_0 + B_0)(CI)_0$$

$$i = 1: (CI)_1 = (CO)_0$$

$$S_1 = A_1 \oplus B_1 \oplus (CO)_0$$

$$= A_1 \oplus B_1 \oplus (A_0 B_0 + (A_0 + B_0)(CI)_0)$$

$$(CO)_1 = A_1 B_1 + (A_1 + B_1)(CO)_0$$

$$= A_1 B_1 + (A_1 + B_1)(A_0 B_0 + (A_0 + B_0)(CI)_0)$$

$$i = 2: (CI)_2 = (CO)_1$$

$$= A_1 B_1 + (A_1 + B_1)(A_0 B_0 + (A_0 + B_0)(CI)_0)$$

$$(CO)_2 = A_2 B_2 + (A_2 + B_2)(CI)_2$$

$$= A_2 B_2 + (A_2 + B_2)(A_1 B_1 + (A_1 + B_1)(A_0 B_0 + (A_0 + B_0)(CI)_0))$$

$$S_2 = A_2 \oplus B_2 \oplus (CI)_2$$

$$= A_2 \oplus B_2 \oplus (A_1 B_1 + (A_1 + B_1)(A_0 B_0 + (A_0 + B_0)(CI)_0))$$

$$\vdots$$

三、用加法器设计组合电路

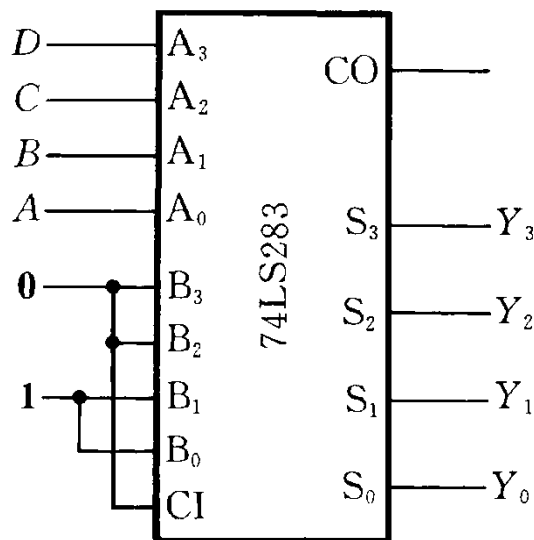
- 基本原理：

若能生成函数可变换成输入变量与输入变量相加

若能生成函数可变换成输入变量与常量相加

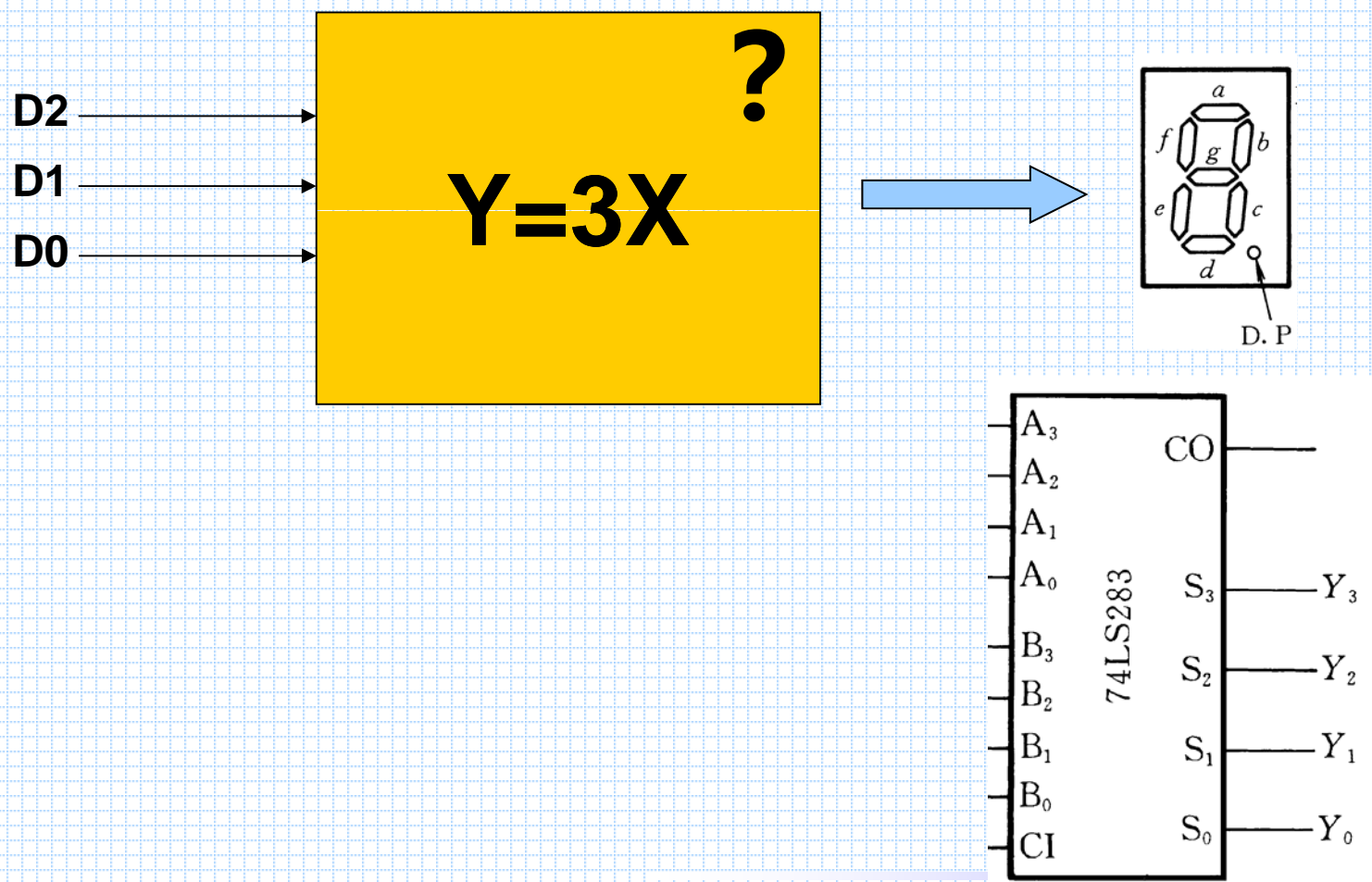
例：将BCD的8421码转换为余3码

$$Y_3 Y_2 Y_1 Y_0 = DCBA + 0011$$



输 入				输 出			
D	C	B	A	Y ₃	Y ₂	Y ₁	Y ₀
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

思考：已知 X 是3位二进制数（其值小于等于5），试实现 $Y=3X$ 并用7段数码管进行显示？



4.3.5 数值比较器

- 用来比较两个二进制数的数值大小

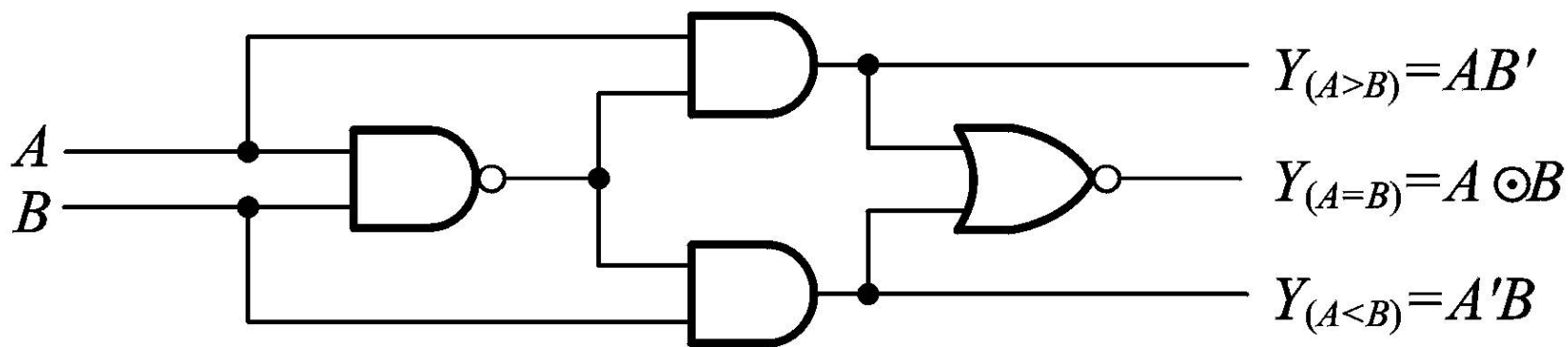
一、1位数值比较器

A, B 比较有三种可能结果

* $A > B (A = 1, B = 0)$ 则 $AB' = 1, \therefore Y_{(A>B)} = AB'$

* $A < B (A = 0, B = 1)$ 则 $A'B = 1, \therefore Y_{(A<B)} = A'B$

* $A = B (A, B \text{ 同为 } 0 \text{ 或 } 1)$, $\therefore Y_{(A=B)} = (A \oplus B)'$



二、多位数值比较器

1. 原理：从高位比起，只有高位相等，才比较下一位。

例如：

比较 $A_3A_2A_1A_0$ 和 $B_3B_2B_1B_0$

$$Y_{(A<B)} = A_3'B_3 + (A_3 \oplus B_3)' A_2'B_2 + (A_3 \oplus B_3)' (A_2 \oplus B_2)' A_1'B_1 \\ + (A_3 \oplus B_3)' (A_2 \oplus B_2)' (A_1 \oplus B_1)' A_0'B_0$$

$$Y_{(A=B)} = (A_3 \oplus B_3)' (A_2 \oplus B_2)' (A_1 \oplus B_1)' (A_0 \oplus B_0)'$$

$$Y_{(A>B)} = (Y_{(A<B)} + Y_{(A=B)})'$$

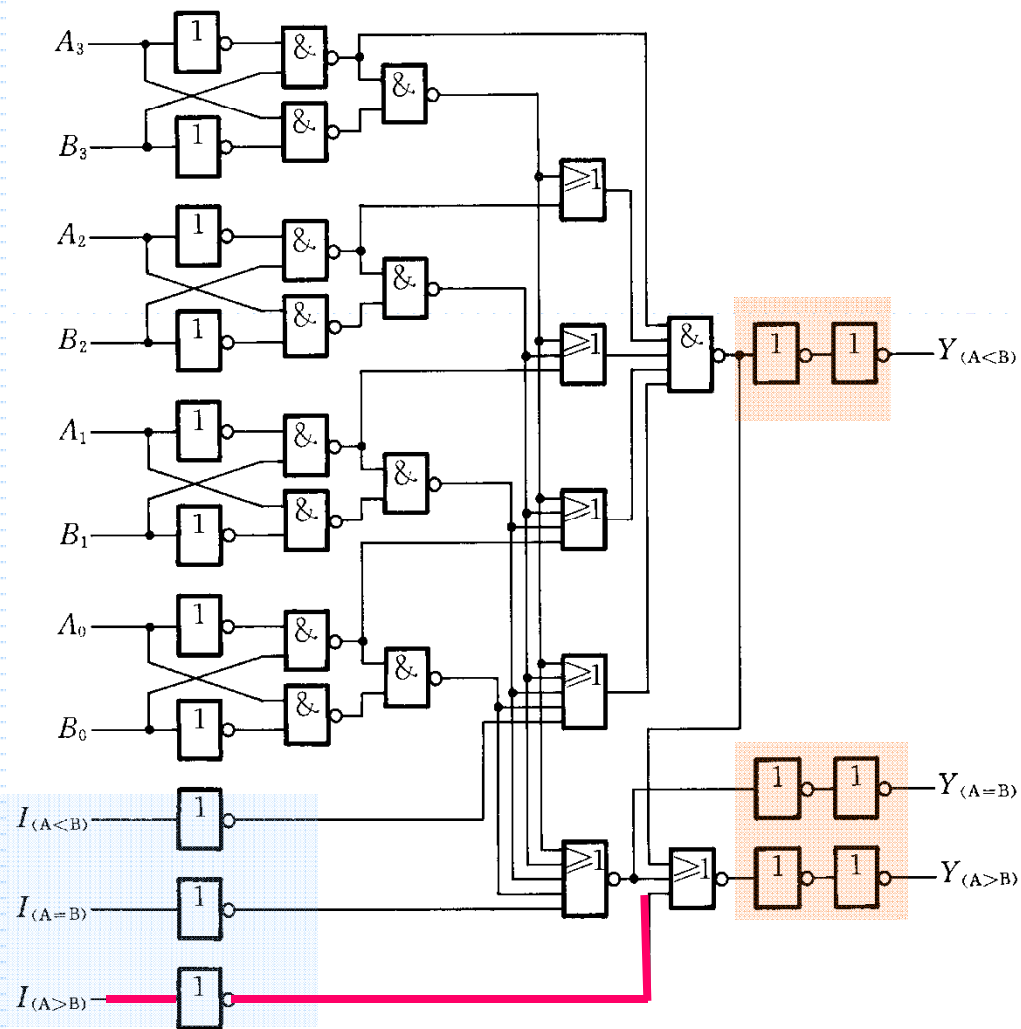
2. 集成电路CC14585 实现4位二进制数的比较

$I_{(A<B)}$, $I_{(A=B)}$ 和 $I_{(A>B)}$ 为附加端, 用于扩展

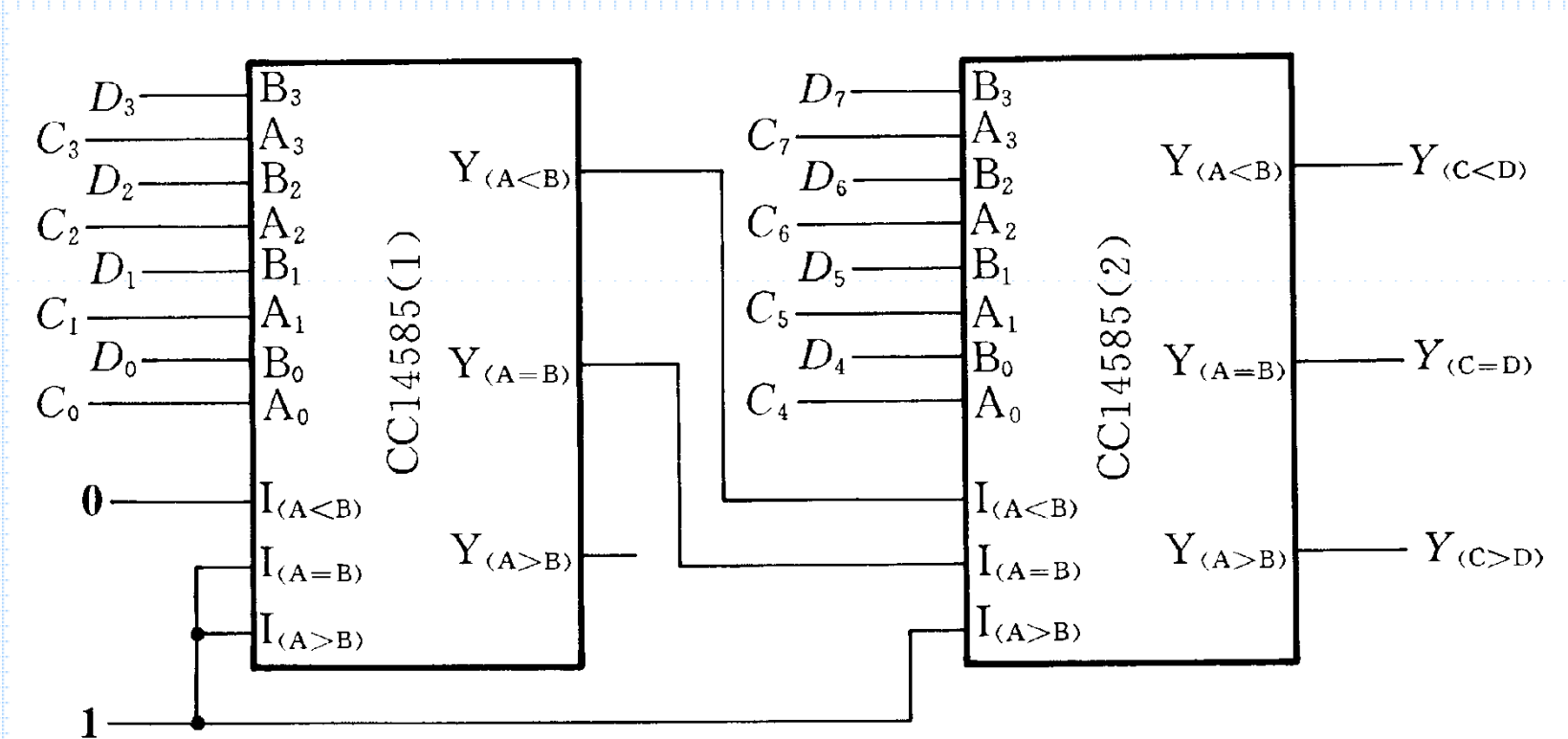
$I_{(A<B)}$, 来自低位的比较结果

$I_{(A=B)}$, 来自低位的比较结果

$I_{(A>B)}$, $A > B$ 输出允许信号



3. 比较两个8位二进制数的大小

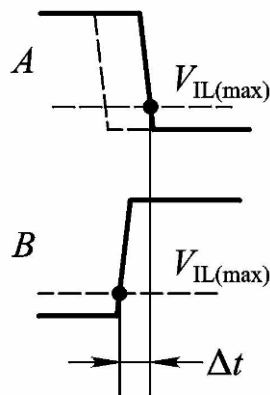


4.4 组合逻辑电路中的竞争-冒险现象

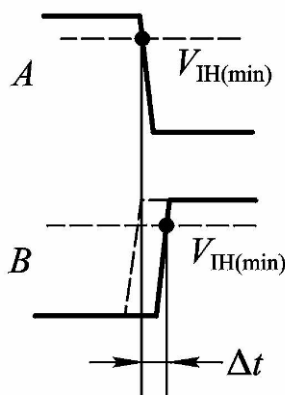
竞争-冒险现象及成因

一、什么是“竞争”

两个输入“同时向相反的逻辑电平变化”，称存在“竞争”



(a)



(b)

二、因“竞争”而可能在输出产生尖峰脉冲的现象，称为“竞争-冒险”。

消除竞争-冒险现象的方法

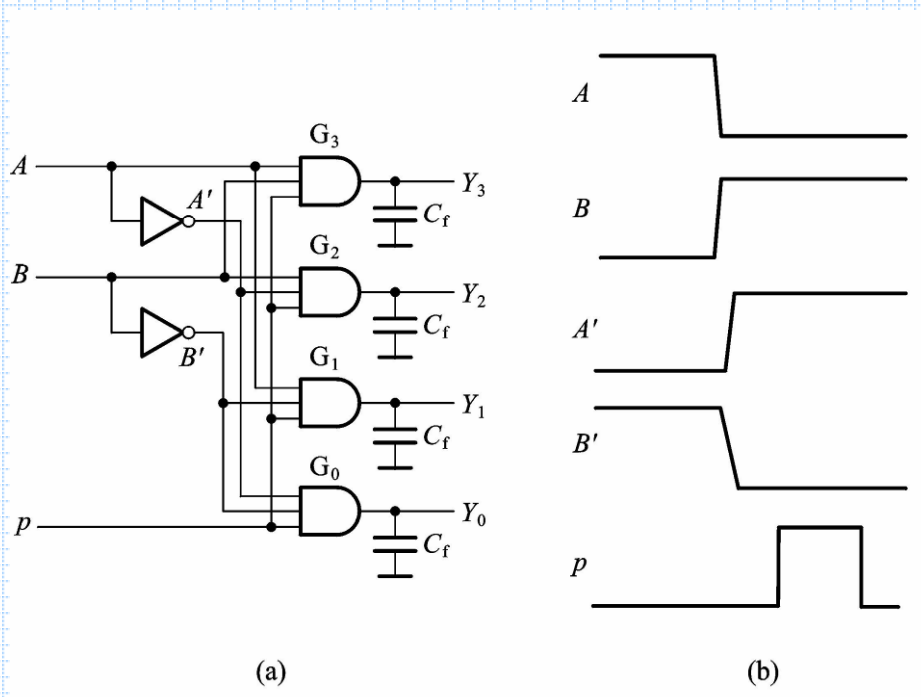
一、接入滤波电容

尖峰脉冲很窄，用很小的电容就可将尖峰削弱到 V_{TH} 以下。

二、引入选通脉冲

取选通脉冲作用时间，在电路达到稳定之后， P 的高电平期的输出信号不会出现尖峰。

三、修改逻辑设计。

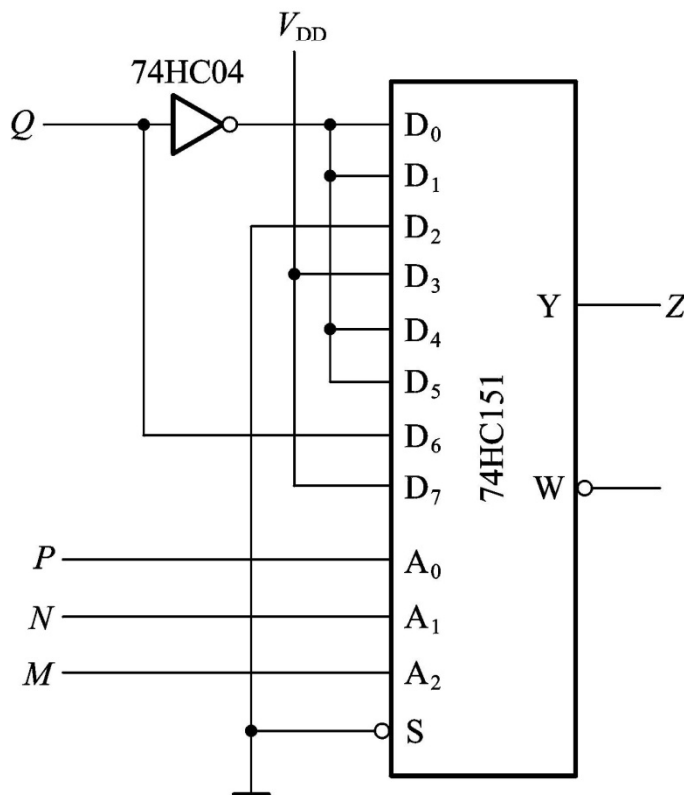


4.5 层次化和模块化的设计方法

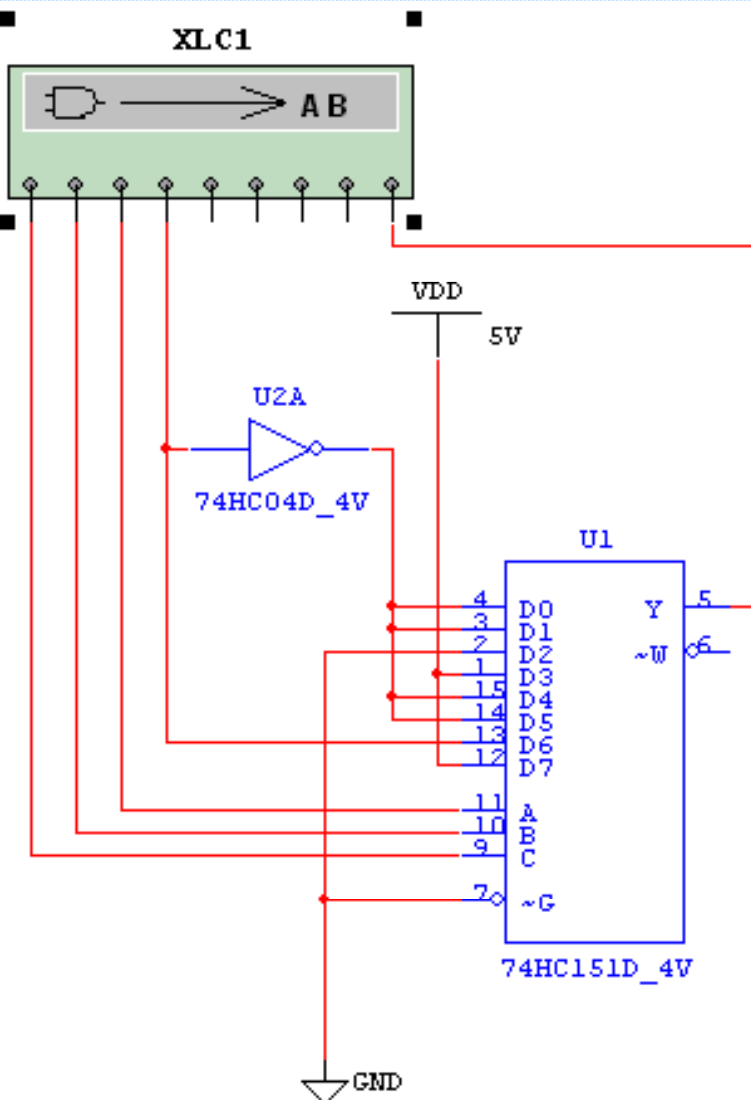
- 由较少的位组合成较多的位
- 由简单模块组合实现复杂功能
- 层次化和模块化意味着可以低成本复制，快速提高性能
- 自顶向下（top-down）和自底向上（bottom-up）

4.6 用multisim分析组合逻辑电路

例:用multisim分析逻辑电路.找出电路的逻辑函数式和逻辑真值表。



4-5-1



Logic Converter-XLC1

	A	B	C	D	E	F	G	H	Out
000	0	0	0	0					1
001	0	0	0	1					0
002	0	0	1	0					1
003	0	0	1	1					0
004	0	1	0	0					0
005	0	1	0	1					0
006	0	1	1	0					1
007	0	1	1	1					1
008	1	0	0	0					1
009	1	0	0	1					0
010	1	0	1	0					1
011	1	0	1	1					0
012	1	1	0	0					0
013	1	1	0	1					1
014	1	1	1	0					1
015	1	1	1	1					1

Conversions

\Rightarrow \rightarrow $\overline{1}0\overline{1}$
 $\overline{1}0\overline{1}$ \rightarrow $A\overline{B}$
 $\overline{1}0\overline{1}$ $\xrightarrow{\text{SIMP}}$ $A\overline{B}$
 $A\overline{B}$ \rightarrow $\overline{1}0\overline{1}$
 $A\overline{B}$ \rightarrow \Rightarrow
 $A\overline{B}$ \rightarrow NAND

$B'D' + ABD + BC$